Hierarchical Novelty Detection

Paolo Simeone[1], Raúl Santos-Rodríguez[2], Matt McVicar[3], Jefrey Lijffijt[1], and Tijl De Bie[1]

[1] Ghent University – IDLab
Department of Electronics and Information Systems, Belgium
{paolo.simeone,jefrey.lijffijt,tijl.debie}@ugent.be
[2] University of Bristol, Department of Engineering Mathematics, UK
enrsr@bristol.ac.uk,
[3] Jukedeck Tech Hub, 20 Ropemaker St, London EC2Y 9AR, UK
mattjamesmcvicar@gmail.com

Abstract. Hierarchical classification is commonly defined as multi-class classification where the classes are hierarchically nested. Many practical hierarchical classification problems also share features with multi-label classification (i.e., each data point can have any number of labels, even non-hierarchically related) and novelty detection (i.e., some data points are novelties at some level of the hierarchy). A further complication is that it is common for training data to be incompletely labelled, e.g. the most specific labels are not always provided. In music genre classification for example, there are numerous music genres (multi-class) which are hierarchically related. Songs can belong to different (even non-nested) genres (multi-label), and a song labelled as `Rock` may not belong to any of its sub-genres, such that it is a novelty *within* this genre (novelty-detection). Finally, the training data may label a song as `Rock` whereas it really could be labelled correctly as the more specific genre `Blues Rock`. In this paper we develop a new method for hierarchical classification that naturally accommodates every one of these properties. To achieve this we develop a novel approach, modelling it as a Hierarchical Novelty Detection problem that can be trained through a single convex second-order cone programming problem. This contrasts with most existing approaches that typically require a model to be trained for each layer or internal node in the label hierarchy. Empirical results on a music genre classification problem are reported, comparing with a state-of-the-art method as well as simple benchmarks.

Keywords: Hierarchical Classification, Novelty Detection, Optimization, Music Genre Classification, Music Information Retrieval

1 Introduction

Multi-Class Classification (MCC) is defined as the task of assigning one of three or more labels to a training instance. Most approaches to MCC break this problem down in a set of simpler problems, typically two-class classification problems. Examples include One vs All, One vs One and Error Correcting Output Code

(ECOC) approaches [1]. These approaches, however, fail to take into account particular structure amongst the labels. A particular case of interest is Hierarchical Classification (HC), where labels are hierarchically nested. For example in Music Genre Classification (MSC), if a song is labelled with `Blues Rock` the hierarchical relation among genre labels implies that it can be labelled with `Rock` as well. Also in domains such as image [2], text [17] and phoneme [5] classification, assuming hierarchical label relations provides both enhanced accuracy and more straightforward interpretation.

As a motivation for the present paper, we use the Music Genre Classification (MGC) task [4, 10, 11, 13, 15]. MGC is confronted with a number of issues:

1. Music genres are hierarchically organised (see Sec. 2.1 for a discussion). This means that if a label applies (e.g. `Blues Rock`), then all more generic labels must automatically apply as well (e.g. `Rock`).
2. Some songs are influenced by several non-hierarchically related genres. Thus, several labels can apply, even if not directly related in the genre hierarchy.
3. Some songs can only sensibly be categorised into high-level genres (commonly cited examples: `Rock`, `Classical`) and not easily into any subgenre, whereas others might be highly specific (`Melodic Death Metal`).
4. However, the labels provided for some data points in commonly available training sets are not always maximally specific, i.e., a Blues Rock song may be labelled as `Rock` but not as `Blues Rock`. Some genre labels may also be altogether missing, e.g. because the genre label was unknown by the data annotator. In other words, while the *presence* of a label in the training implies that this label applies to that song, the *absence* of a label does not imply that it does not apply.

The first issue makes MGC a HC problem. Due to the second issue, MGC is also inherently a Multi-Label Classification problem. The third issue means that for each class there are really only positive examples available in the training set, as the absence of a label for a data point does not imply that that data point is a negative example for that class. MGC is thus reminiscent of Novelty Detection (ND) [14]. Finally, to account for the fourth issue, a suitable method must be capable of predicting a (set of) label(s) that are not necessarily maximally specific. We believe that these four issues are relevant in many practical problems beyond MGC.

A naive approach to tackle issues 2-4 would be train a ND method for each of the different classes. Given a song, it can then be labelled with the genres for which the song is not a novelty. Unfortunately, this approach neglects the hierarchical nature of the set of genre labels (issue 1).

In this paper we are interested in extending a well-known ND method to the hierarchical setting, leading to a method that we call Hierarchical Novelty Detection (HND). To the best of our knowledge, HND is the first method that accounts for *all four issues* highlighted above. A further benefit of HND is that it can be trained by solving a single convex second-order cone programming problem, making it a global or 'big bang' approach—such approaches are considered preferable over the more common approaches that train a collection of models

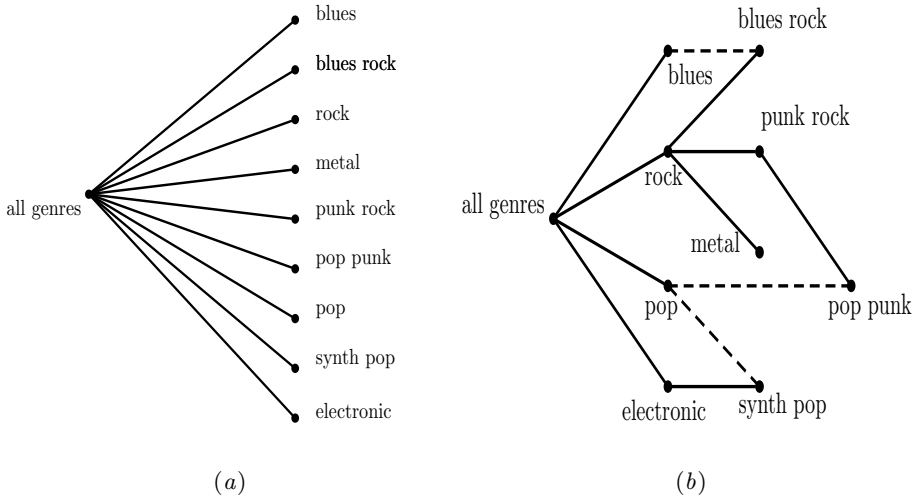

Fig. 1. Example of music genre relationships. (a): flat classification, where every genre is a subgenre of the root. (b) tree structure, where the hierarchical nature of genres is explicit. If links such as `Blues` → `Blues Rock` are included, the tree becomes a DAG.

for different levels in the hierarchy of labels [13]. Additionally the method allows for labels to belong to various branches. Whether a novelty or not, we are able to enrich the information about data points and this is relevant in MGC at least for automatic identification of erroneously-annotated genre labels and for identification of emerging music styles. To validate the method, we present experiments on a set of samples from a publicly available dataset tagged by `Last.fm` users. We defined a hierarchy over the genres in these data, but intentionally left a subset out, to assess the extent to which our algorithm would identify songs which belonged to this missing node in the graph as being novel.

The remainder of this paper is structured as follows. In Section 2 we review existing approaches before introducing our proposed method in Section 3. Experiments and conclusions are presented in Section 4.

2 Background

In this Section we delve deeper into how hierarchical classification and novelty detection operate and outline the terminology and notation using MGC as domain, so that we can introduce our method concisely in Section 3. We begin with a discussion on how relationships between genres can be represented mathematically.

2.1 Genres relationships: graphs, DAGs or trees?

For many problems, as in MGC, the relationship between classes can be formalised mathematically as a graph, with nodes equal to genres and edges be-

tween nodes representing relationships between genres. The exact nature of the graph is the subject of current scholarly work. The model could be a directed or undirected graph, with or without weights on the edges, with or without cycles (in the directed case). Commonly the choice is between trees or Direct Acyclic Graphs (DAG), but any election of a graph structure is only an approximation of reality, a compromise reached for pure modelling purposes. Here, the set of music genres is modelled as a Directed Acyclic Graph (DAG); see Fig. 1(*b*). The *root* in this DAG will correspond to *all genres*, while the *leaf nodes* correspond to the *most specific genres*. An edge connecting genre A to genre B means that genre B is a subgenre of genre A. The flexibility of a DAG structure means that, for example, `Blues Rock` may be a subgenre of both `Blues` and `Rock`.

2.2 Hierarchical classification

A very naive approach is a single multi-class classifier trained to discriminate between each of the classes. This model ignores the hierarchical structure of the labels given by the nature of music genres and therefore often performs poorly when compared to models that explicitly exploit this information. The three main hierarchical approaches are listed below. For a comprehensive review of HC we refer the reader to [13] and the references therein.

Flat classifier approach A flat classification approach ignores the hierarchy being able to predict only the leaf nodes, i.e., the most specific subclasses, and then considers the *IS-A* relationship in the hierarchy for a multi-label classification. Such an approach is unlikely to be beneficial for the performance of the classifier.

Local classifier approach These methods are designed as a sequence of flat classifiers. They are often top-down in nature, first classifying each test point into one of the children of the root, and then iterating over the children. Either the process continues until the test point reaches a leaf (*mandatory leaf node prediction* in the language of [13]), or some stopping criteria are applied (*non-mandatory leaf node prediction*).

Global classifier approach The most sophisticated approaches yield a single overall model, trained at once on the entirety of the data. These are known as 'big-bang' approaches and are the most principled. However, only a handful of methods exist that fall into this category [7, 9, 12]. Our method falls within this category.

2.3 Novelty detection

The goal of Novelty Detection (ND) is to decide if new points belong to (one or more) classes present in the data. A natural way to approach this problem is to enclose the data within a decision surface, with any points that fall outside the surface classified as novel. The most common surface is the hypersphere [14].

Hard margin novelty detection Let $\mathbf{x} = \{x_1, \ldots, x_n\}$ be a set of n data points with $x_i \in \mathbb{R}^d, i = 1, \ldots, n$. The simplest form of novelty detection fits a hypersphere $\mathcal{H} = (\mu, R)$ around the data, specified by a centre vector $\mu \in \mathbb{R}^d$ and radius $R \in \mathbb{R}$. Finding the $\mathcal{H}$ which most tightly fits the data $\mathbf{x}$ amounts to solving the following optimisation problem:

$$\min_{R,\mu} \; R^2, \quad \text{subject to} \tag{1}$$

$$R \geq 0 \tag{2}$$

$$||\mathbf{x}_i - \mu||^2 \leq R^2, \quad i = 1, \ldots, n, \tag{3}$$

where $||\cdot||$ represents Euclidean norm. The objective (3) is convex in R^2 and μ, meaning the above problem can be solved efficiently using gradient descent or similar methods.

Soft margin novelty detection A common variant of the problem above is to allow some points to be slightly outside the enclosing hypersphere. This allows for extreme values to be ignored and is more robust in train/test settings. Introducing slack variables $\xi \in \mathbb{R}, i = 1, \ldots, n$ and a hyper-parameter $C \in \mathbb{R}$, the objective then becomes:

$$\min_{R,\mu,\xi} \; R^2 + C \sum_{i=1}^{n} \xi_i, \quad \text{subject to} \tag{4}$$

$$\xi_i \geq 0, \quad R \geq 0 \tag{5}$$

$$||\mathbf{x}_i - \mu||^2 \leq R^2 + \xi_i, \quad i = 1, \ldots, n, \tag{6}$$

This problem is still convex, but has an increase in computational complexity due to the increased number of parameters.

Soft margin novelty detection with multiple classes Similarly to the multiple one-class SVM-based method [8], it is possible to perform novelty detection with multiple known classes. One simply solves the problem (4) subject to (6) for each class. Suppose that for each of the x_i we have a label $y_i \in \{1, \ldots, K\}$. Then it is required to find K hyperspheres $\{\mathcal{H}_1, \ldots, \mathcal{H}_k\} = \{(\mu_1, R_1), \ldots, (\mu_K, R_K)\}$, which may be done via:

$$\min_{R,\mu,\xi} \; \sum_{i=1}^{K} R_k^2 + C \sum_{i=1}^{n} \xi_i, \quad \text{subject to} \tag{7}$$

$$\xi_i \geq 0, \quad R_k \geq 0 \tag{8}$$

$$||\mathbf{x}_i - \mu_{y_i}||^2 \leq R_{y_i}^2 + \xi_i, \quad i = 1, \ldots, n. \tag{9}$$

Note that since there is no interaction between any hypersphere μ_k or R_k, this problem can be solved independently for each k. However, using the above formulation is faster in practice due to the reduced overhead.

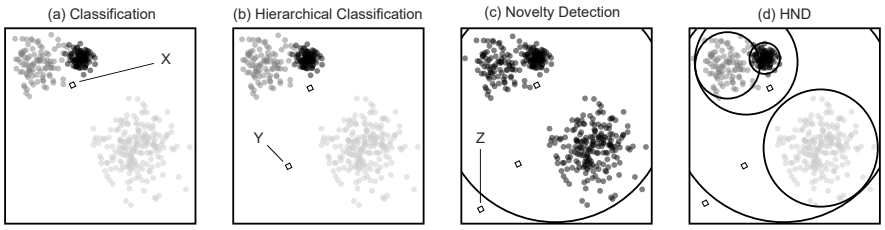

Fig. 2. Flat classification, HC, ND and the proposed HND within a three classes example: `black`, `grey` (upper left corner) and `light grey` (lower right corner). In (a), the point x is classified as belonging to the `grey` class. In (b) and subsequent plots, `grey` and `black` are assumed to share the same parent. Here, x is classified as `grey_or_black` and then possibly into one of `grey` or `black`, and y is classified as `light grey`. In (c), x and y are classified as `normal` whilst z is classified as `novel`. Finally in (d), x is classified as `grey_or_black`, but novel to the classes `grey`, `black` and `light grey`, whilst y is classified to be belonging to the dataset—see (c)—but none of its subclasses. z is considered novel as before. Note that for (*c*) and (*d*) the slack variables allow some points to fall slightly outside the spheres.

Using 2-dimensional toy data, Fig. 2 illustrates the differences between flat classification, hierarchical classification, novelty detection and our proposed hierarchical novelty detection, that we introduce below.

3 Hierarchical Novelty Detection and Classification

As previously illustrated in Sec. 1, hierarchical classification methods must satisfy a number of design criteria. They must be able to operate in the *multi-label setting* (both for the training set and for prediction). They must be able to work with *hierarchies that can be DAGs*, not just trees. They must predict label sets that are *upper sets* of the DAG: for each predicted genre, all of its more generic genres (higher in the DAG) must be predicted as well. We will refer to this criterion as the *consistency criterion* of the label set. And finally, they must be able to deal with training examples for which the *label sets are incomplete*. To the best of our knowledge, no existing methods satisfy all of the design criteria listed above and, importantly, none of them naturally deals with the fact that the training label sets may be incomplete.

3.1 Hierarchical Novelty Detection

We take inspiration from the novelty detection methods listed in Sec. 2.3, with the only addition that we enforce relationships between the hyperspheres ($\mathcal{H}_1$, ..., $\mathcal{H}_K$) by encoding the dependencies present in the hierarchy. Suppose that the labels y_i are arranged in a *hierarchy*, which may be realised as a rooted DAG with nodes labelled $\{1, \ldots, K\}$. Assuming w.l.o.g. that the label 1 is the

root of the DAG, each label $k \in \{2, \ldots, K\}$ has a *parent* label set, which we denote by $\mathrm{Pa}(k)$. We then represent the subclass constraint by insisting that the hyperspheres for the subclasses are nested:

$$\min_{R,\mu,\xi} \sum_{i=1}^{K} R_k + C \sum_{i=1}^{n} \xi_i, \quad \text{subject to} \tag{10}$$

$$\xi_i \geq 0, \quad R_k \geq 0 \tag{11}$$

$$||\mathbf{x}_i - \mu_{y_i}|| \leq R_{y_i} + \xi_i, \quad i = 1, \ldots, n. \tag{12}$$

$$||\mu_{\mathrm{Pa}(k)} - \mu_k|| \leq R_{\mathrm{Pa}(k)} - R_k, \quad k = 2, \ldots, K, \tag{13}$$

This is a convex problem and is always feasible as we can trivially set all μ_k to be the centre of the data, R_k equal to each other and large enough to encapsulate all the data, and ξ_i equal to 0. It is clear that in this case Equations (11–13) are satisfied, and since the problem is convex we may use this as an initial solution and improve our objective to a global minimum. It also admits an interpretation of the C parameter, as we demonstrate below.

3.2 Interpretation of the hyperparameter C

Along with convexity, Equations (10–13) present another interesting property regarding the interpretation of the hyperparameter C. Suppose that (10) is solved optimally for a given set of data points, and that the optimal objective is found to be $L^* = \sum_{k=1}^{k} R_k^* + C \sum_{i=1}^{n} \xi_i^*$. Now suppose that there is an increase in all of the R_k^*, so that $R_k^* \to R_k^* + \Delta$, where $\Delta \in \mathbb{R}^+$. Adding Δ to each of the K radii corresponds to adding $K\Delta$ to the objective and, since L^* is optimal, this means there must be a reduction in $C \sum_{i=1}^{n} \xi_i^*$ of $K\Delta$. Therefore we want to determine how the slacks ξ_i^* adjust in order to meet this arbitrage. With the original values of R^*, ξ_i^* satisfied,

$$\xi_i^* \geq ||x_i - \mu_{y_i}^*|| - R_{y_i}^* \tag{14}$$

by simply re-arranging constraint in (12). Denoting the adjusted slack variables as $\hat{\xi}_i$, for values of the new R^*, they must satisfy,

$$\hat{\xi}_i \geq ||x_i - \mu_{y_i}^*|| - R_{y_i}^* - \Delta. \tag{15}$$

Taking the difference of (14) and (15), summing over i, and multiplying by C we arrive at,

$$C \sum_{i=1}^{n} \xi_i^* - \hat{\xi}_i \geq Cn\Delta. \tag{16}$$

However, note that for some of the x_i, the slacks will not need to be adjusted, as they will remain within $R_{y_i}^*$ of μ_{y_i}. Let $\mathcal{J} \subseteq \{1, \ldots, n\}$ be the set of points for which the slacks need to be adjusted, with $|\mathcal{J}| = J \leq n$. J can be understood as the number of *outliers* in the optimal solution, as it corresponds to the number

of points which are outside their corresponding hypersphere. For all other $i \notin \mathcal{J}$ we may set $\hat{\xi}_i = \xi_i^*$ and so (16) becomes,

$$C \sum_{j \in \mathcal{J}} \xi_j^* - \hat{\xi}_j \geq CJ\Delta. \tag{17}$$

Recall that we require this change to be $K\Delta$ at the optimum, so we conclude that $CJ\Delta \geq K\Delta$ or equivalently $C \geq \frac{K}{J}$. Remarkably, this means that we may adjust C to directly control how many outliers we are willing to tolerate in our problem. For example, if we wish to tolerate exactly one outlier, we should set $C = K$.

4 Experimental Results

To validate our method we considered a comparison, using Hierarchical Precision and Recall as defined in [13], against our implementations of the methods depicted in Sec. 2.2 as well as the hierarchical clustering algorithm from CLUS library [16]. The former were implemented using one vs all approach with linear Support Vector Machines as base classifier: a general one discriminating between each class, a flat and a local hierarchical classifier. For the local classifier we used a 'siblings policy' [6] for every node, i.e., considering samples for a parent node and the child siblings as labels. Experiments were run over a subset of the samples provided by the Million Song Dataset [3] selected according to a user defined taxonomy meant to allow for the possibility of discovering novelty genres in the dataset. We will first proceed with the definition of the performance measures and then with a description of the dataset before the final discussion.

4.1 Hierarchical Precision and Recall

Let us consider $\hat{P}_i$ as one element in the set of predicted values for a sample by any of the classifiers and $\hat{T}_i$ as the set of real labels for the same sample. The definitions for hierarchical precision (hP) and recall (hR) are:

$$hP = \frac{\sum_i |\hat{P}_i \cap \hat{T}_i|}{\sum_i |\hat{P}_i|}, \quad hR = \frac{\sum_i |\hat{P}_i \cap \hat{T}_i|}{\sum_i |\hat{T}_i|} \tag{18}$$

It is worth noting how this definition is the most inclusive possible as it considers the possibility of multiple outputs for the classifier, although flat and local solutions will mostly output a value in the hierarchy while considering the superclasses labels. This is a well known problem affecting different methods thoroughly discussed in [13]. In this sense, CLUS is among all methods the most similar to HND for its ability to produce multiple labels along different branches.

4.2 Million Song Dataset

The Million Song Dataset [3] is a publicly available dataset constituted by features collected by the Echo Nest, which has been additionally enhanced by a

Table 1. Summary of the dataset. Columns from left to right indicate respectively name of the main tag/class for a sample, number of samples included in the dataset, maximum number of tags per sample, average number of tags per sample and whether the current class was meant to be discovered as a novelty.

label	samples	max tags	avg tags	novelty
blues rock	100	4	2.07	No
blues	100	5	1.64	No
classical	100	3	1.05	**Yes**
country	100	5	1.46	**Yes**
electronic	100	4	1.58	No
folk	100	6	1.69	**Yes**
jazz	100	5	1.37	**Yes**
metal	100	4	1.33	No
pop punk	100	6	1.89	No
pop	100	4	1.58	No
punk rock	100	5	1.69	No
rock	100	4	1.56	No
synth pop	100	4	1.67	No
all genres	1300	6	1.58	No

collection of annotations retrieved from the `Last.fm` website by means of tags (labels) for every song in the set. These tags were used to select a subset of samples according to a user defined hierarchy of 13 different music genres. However only 9 of them are featured in the taxonomy represented by the thick lines in Fig. 1(b). For the purpose of testing, the algorithm was required to discover the remaining 4 genres as novelties by labelling them as `all genres` samples.

The selection considers all the available tags for a single song. For example a sample tagged as both `rock` and `pop rock` by the users was considered as belonging to both classes. No limit over the number of labels was imposed except for the fact that only labels belonging to the taxonomy were considered. The choices resulted in an expansion of both training and test set: the previously described example was used as training sample for both the classes in such a context. For this reason the set-up affected and in some way penalized the performance according to the given definition of hP and hR because a prediction over the same example was considered partially or fully correct depending on the number of correct tags in the output. Table 1 gives an overview of the sample distribution. Training and test sets were built by splitting the collection in half.

4.3 Discussion and Conclusions

Experiments were performed to first study the effect of the variation of C on the performance in terms of hP and hR. The chosen range for the fraction of novelties was $[0.005, 0.7]$. As shown in Fig. 3(a) HND scored better, as expected, than a traditional method as a general One vs All (OVA) multi-class classifier,

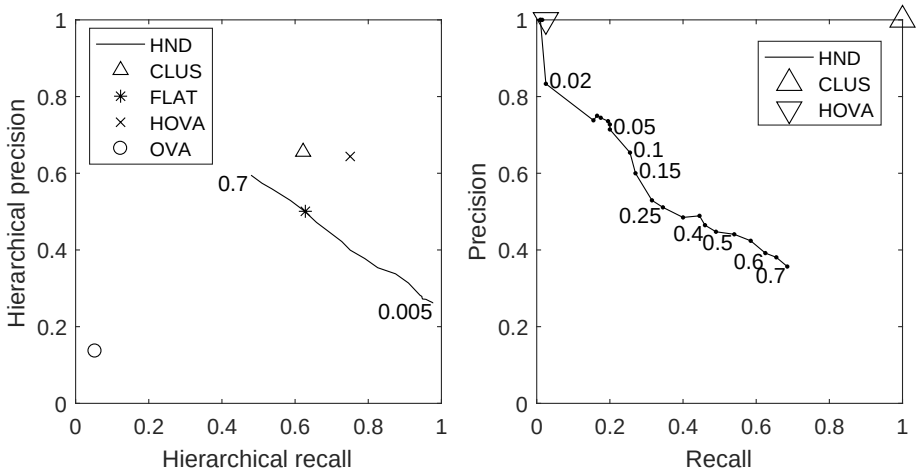

Fig. 3. Evaluation of precision and recall with significant values of C specified along HND plot: (a) Hierarchical precision and recall for all evaluated methods. (b) Precision and recall for novelty detection.

similar to the FLAT classifier for a certain range of C and it was outperformed by other hierarchical strategies like CLUS and the local classifier (HOVA). We believe that in the case of CLUS, better performances are due to the specific clustering strategy which applies discriminant weights while descending down the hierarchy. Unfortunately, our formulation does not allow a straightforward integration (with relative interpretation) of a weighting strategy which might reduce the performance gap.

By definition HOVA and FLAT are able to output multi-class labels, but they are limited to the set of node labels that are on the path between the root and the lowest classified node, but still HOVA is comparable with CLUS. CLUS and HND, especially the latter, may *de facto* be penalized by their ability to descend through different branches of the hierarchy and defected by excessive specialization. The measures defined in (18) may not be adequate to actually evaluate our algorithm.

Hierarchies and their arbitrariness usually represent the weakness of hierarchical approaches as there is no unifying framework for the taxonomies. For instance, consider a sample which is simply labelled `pop punk` in the hierarchy of Fig. 1(b). It is more likely that it belongs to `pop` even if it is just labelled as `pop punk` by many users. In Fig. 3(a) we explicitly marked the values of C close to our method's plot in order to show how increasing the 'discovery' factor affects our precision as the denominator in hP is raised a lot by the tendency to predict too many labels. Conversely high values of recall are due to this tendency of hyperspheres from apparently far genre nodes in the hierarchy being instead very super-imposed. Such observations also suggest how a different model, e.g.

ellipsoids, or a kernel version could lead to further improvements. This will be explored as future work together with the possibility for samples to be labelled as external to the set of nested hyperspheres, which may imply that a genre do not even belong to the range included in `all genres`.

The ability to discover novelties is presented in Fig. 3(b). Both precision and recall were evaluated by considering novelty and non-novelty samples as positive and negative samples, respectively. As expected C varies according to the fraction of novelties and therefore to recall values. However here CLUS shows how its precision may be very high on genres which are effectively far from those belonging to hierarchy. Being those the majority of the test samples, and given the effects of the weighting strategy on favouring higher levels of the hierarchy, the performance values were definitely raised.

Acknowledgments The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement no. 615517, from the EPSRC (EP/M000060/1), from the FWO (project no. G091017N, G0F9816N), and from the European Union's Horizon 2020 research and innovation programme and the FWO under the Marie Sklodowska-Curie Grant Agreement no. 665501.

References

1. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: A unifying approach for margin classifiers. Journal of Machine Learning Research 1, 113–141 (2000)
2. Barutcuoglu, Z., DeCoro, C.: Hierarchical shape classification using Bayesian aggregation. In: Proc. of the IEEE International Conference on Shape Modeling (2006)
3. Bertin-Mahieux, T., Ellis, D.P., Whitman, B., Lamere, P.: The million song dataset. In: Proceedings of the 12th International Conference on Music Information Retrieval (2011)
4. DeCoro, C., Barutcuoglu, Z., Fiebrink, R.: Bayesian aggregation for hierarchical genre classification. In: ISMIR. pp. 77–80 (2007)
5. Dekel, O., Keshet, J., Singer, Y.: An online algorithm for hierarchical phoneme classification. In: Bengio, S., Bourlard, H. (eds.) Proceedings of the First international conference on Machine Learning for Multimodal Interaction. vol. 3361, pp. 146–158. Springer (2004)
6. Fagni, T., Sebastiani, F.: On the selection of negative examples for hierarchical text categorization. In: Proceedings of the 3rd Language Technology Conference. pp. 24–28 (2007)
7. Garcia-Garcia, D., Santos-Rodriguez, R.: Sphere packing for clustering sets of vectors in feature space. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. pp. 2092–2095. IEEE (2011)
8. Jumutc, V., Suykens, J.A.K.: Multi-class supervised novelty detection. IEEE Trans. Pattern Anal. Mach. Intell. 36(12), 2510–2523 (2014)

9. Kiritchenko, S., Matwin, S., Famili, A.F.: Functional annotation of genes using hierarchical text categorization. In: Proc. of the BioLINK SIG: Linking Literature, Information and Knowledge for Biology (2005)
10. Li, T., Ogihara, M.: Music genre classification with taxonomy. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. vol. 5, pp. v–197 (2005)
11. Pachet, F., Cazaly, D., et al.: A taxonomy of musical genres. In: Content-Based Multimedia Information Access. pp. 1238–1245 (2000)
12. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, New York, NY, USA (2004)
13. Silla Jr, C., Freitas, A.: Novel top-down approaches for hierarchical classification and their application to automatic music genre classification. In: IEEE International Conference on Systems, Man and Cybernetics. pp. 3499–3504. IEEE (2009)
14. Tax, D.M., Duin, R.P.: Support vector data description. Machine Learning 54(1), 45–66 (2004)
15. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE transactions on Speech and Audio Processing 10(5), 293–302 (2002)
16. Vens, C., Struyf, J., Schietgat, L., Dzeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Machine Learning 73(2), 185–214 (2008)
17. Xue, G.R., Xing, D., Yang, Q., Yu, Y.: Deep classification in large-scale text hierarchies. In: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 619–626. New York, NY, USA (2008)